Software Verification Process Improvement Proposal Using Six Sigma

Tihana Galinac[1] and Željka Car[2]

[1] Ericsson Nikola Tesla, Research and Development Center, Krapinska 45, HR-10000 Zagreb, Croatia
tihana.galinac@ericsson.com
[2] University of Zagreb, Faculty of Electrical Engineering and Computing, Unska 3, HR-10000 Zagreb, Croatia
zeljka.car@fer.hr

Abstract. In the rapidly growing modern telecommunications industry, software quality and reliability requirements are in contrast to the shorter time to market and higher complexity requirements dictated by strong competition on the telecommunications market. In such a rapidly changing environment, software development organization must improve almost on a daily basis in order to achieve the operational excellence which leads to business success. In this paper, an approach to the continuous improvement of the software verification process based on the application of Six Sigma is given. More precisely, with the help of the Six Sigma methodology, change management, and statistical tools and techniques, the proposed approach solves the problem of fault slippage through verification phases, which is particularly important in overlapping project conditions. Success of the proposed process improvement, proved using Six Sigma methodologies for a case study from a real industrial project, encourages wide and general application to any software verification process.

1 Introduction

Significant research effort in software development during the last two decades has been devoted to Software Process Improvement (SPI) [15]. It is mostly triggered from industry [2]. The main problems faced by industry include rapid the software development and shorter time to market requirements caused by strong competition and new technologies. Consequently, final products are often of lower quality and development and verification expenses are greater. This effect is especially emphasized in telecommunications software development, due to its specific nature, as will be explained in Sect. 2.

Since software quality is measured with the number of faults, where fewer faults imply significant savings in rework time and cost, SPI is mostly concerned with the verification part of the software development process [14], [13], [1], [4], [8]. This includes not only better fault detection, but also better fault prevention.

J. Münch and P. Abrahamsson (Eds.): PROFES 2007, LNCS 4589, pp. 51–64, 2007.

One of the main goals of software verification improvement is reducing fault slippage through verification phases. It is also the main problem addressed in this paper.

There are several SPI approaches. The International Standards Organization (ISO) has established several standards for management of software quality [11], [12]. Their quality assurance procedure is based on reviews and audits measuring the ability of developers to fulfill the work assignments initially specified. The Capability Maturity Model (CMM) introduces continual improvement on the process management [6], [7]. The idea behind CMM is that independent assessments can be used to grade organizations based on how well they create software according to their definition and execution of their processes.

Nowadays, the importance of statistical methods and mechanisms is growing rapidly [5]. They are becoming very promising candidates for improving process control by reaching higher production efficiency and maturity levels, as well as early indicators of the need for changes in processes. However, due to the complexity of organization, Quality Management goes far beyond only statistics. Having that in mind, the Six Sigma approach [3] is a set of change management skills, and quality and statistical knowledge, based on the Define, Measure, Analyze, Improve, Control (DMAIC) methodology. One of the most important aspects of the Six Sigma approach is its statistical base and statistical process control which forms its integral part.

Originally one of the most successful approaches used in hardware production processes [15], Six Sigma is becoming more and more interesting to the software industry. However, as explained in [2], it is still unclear how to fully apply the concept of Six Sigma to software development. In this paper, we present an improvement proposal and show its successful deployment and further control, obtained as an outcome of the Six Sigma project. The precise research framework is given in Sect. 2.

The paper is organized as follows. In Sect. 2, we give an overview of the software verification process by analyzing the problem and establishing a research framework. Section 3 explains the problem in more detail and provides a cost analysis that was used before improvement project initiation to encourage investment. Furthermore, it deals with project execution, which includes defining, measuring and analyzing the problem, through logically elaborated steps leading to the final solution. Finally, Sect. 4 presents an improvement proposal, describes achieved benefits, provides a strategy for future process control and gives guidelines for implementing improvements for the general case.

2 Research Framework

The case study of this paper was performed for an industrial project within a development unit at Ericsson which aims to achieve business excellence through continuous improvement programs. It deals with developing large scale software for telephone exchanges forming network solutions for next generation networks. The development unit is a multinational organization consisting of four